

# EFFECT OF RIGIDITY ON TRILATERATION TECHNIQUE FOR LOCALIZATION IN WIRELESS SENSOR NETWORKS

Dr. Saroja Kanchi

Department of Computer Science,  
1700 University Blvd,  
Kettering University,  
Flint MI 48504  
skanchi@kettering.edu

## **ABSTRACT**

*The localization of wireless sensor networks is an important problem where the location of wireless sensors is determined using the distance between sensors. Trilateration is a geometric technique used to find location of points in 2D using distances. Using geometry, one can find the location of a point uniquely in 2D given its distance to three other points in 2D. The problem of finding the trilateration order of vertices even if the network of sensors is a uniquely localizable is NP-Complete. The 2D localization problem is closely related to the problem of graph rigidity. A graph can be uniquely realized in 2D if and only if the underlying network graph is globally rigid. Therefore by examining the structure of the underlying graph for rigidity and localization guided by rigidity is another technique used in localization.*

*We study the performance of trilateration which is based on geometry and local information to see if it is effected by graph rigidity which is a global property. In particular, we compare the performance of the trilateration on connected non-rigid networks and connected rigid networks. We focus on sparse networks graphs of lower radius.*

## **1. INTRODUCTION**

The recent advancements in wireless communication and sensing technology have resulted in wide deployment of sensors in applications like environmental monitoring, search and rescue, military surveillance, and intelligent transportation, etc [1, 2, 3]. In these types of applications, the knowledge of the location of each sensor is important. Due to constraints of these application, however it is often difficult to preset the locations of sensors before they are deployed. Therefore, the capability of obtaining the positions of sensors after the deployment is fundamental to the success of the mission of sensor networks. Most of the node localization algorithms are based on range measurements, through either time of arrival (TOA) [4], time difference of arrival (TDOA) [5], or received signal strength (RSS) [6, 7]. The problem of *localization* is to derive the geolocation of a node given a set of known locations and range measurements to these locations. Given the available range measures, if there is only one position for the nodes in the network,

then the network is localizable. Similarly, if a node has only one position that satisfies all the range measures relevant to it, it is localizable.

Yang [8] presented conditions for node localizability using Trilateration technique. Yang et al [9] presented a distributed algorithm for localization which uses an extended trilateration. Wheel graphs are used as basic localizable subgraph and localization then extends to adjacent wheels. Li [10] provided a path for the mobile beacon based on depth-first search and used a variation of trilateration in the DREAMS technique. The geometric technique of trilateration is attractive due to the fact that the algorithms can be implemented as distributed algorithms.

The problem of network localization is closely related to the graph rigidity. A network defined by a set of nodes and a set of known distances between the nodes can be localized only if the graph derived from the network is uniquely realizable. It has been shown [11, 12, 13] that for a graph to be uniquely realizable, it must be redundantly rigid and tri-connected. Jacobs [11] proposed a centralized polynomial algorithm to check the rigidity of a graph through pebble games, trilateration alone. However, these are centralized algorithms. The pebble game algorithm described in Section 2 is based on depth first search and therefore is not distributed. However, distributed algorithms are useful considering the nature in which the sensor networks are deployed in practical setting. Very often there will not be a central node that all nodes in the network can communicate with. Moreover in cases where nodes may move and find new distances to their neighbors, these new distances need not be updated to central location if a distributed algorithm is used for rigidity finding and hence localization. However rigidity is a global property of a graph and it is quite a challenge to check the rigidity property without a huge message complexity.

In this paper we study if Trilateration performs better on rigid graphs versus random graph. Note that if a graph is 6-connected it is globally rigid and therefore localizable. We deal with graphs of low radius and low connectivity to examine if the Trilateration performs better in graphs that are weakly rigid.

## 2. GRAPH RIGIDITY AND ITS RELATION TO WSN LOCALIZATION

In this section, we are going to introduce the theory in network localizability and rigidity. A detailed description can be found in [11, 12, 13]. Let a framework  $p(G)$  be a graph  $G$  along with a mapping  $p : V \rightarrow \mathbb{R}^2$  which assigns each vertex to a point in the plane. A *finite flexing* of a framework  $p(G)$  is a family of realizations of  $G$ , parameterized by  $t$  so that the location,  $r_i$ , of each vertex  $i$ , is a differentiable function of  $t$  and  $|r_i(t) - r_j(t)|^2$  is constant for every  $(i, j) \in E$ . Thinking of  $t$  as time, and differentiating the edge length constraints, we have

$$(u_i - u_j)(r_i - r_j) = 0 \text{ for every } (i, j) \in E \quad (1)$$

An assignment of velocities that satisfies Eq. 1 for a particular framework is an infinitesimal motion of that framework. Every framework has three trivial infinitesimal motion: two translations, and a rotation. If a framework has a nontrivial infinitesimal motion it is infinitesimally flexible. Otherwise it is infinitesimally rigid. Checking for whether a particular framework is rigid or not, can be determined from the property of the graph.

Let  $G = \{V, E\}$  denote a network of vertices  $V = \{1, 2, \dots, n\}$  and for any edge  $(i, j) \in E$ , the distance between  $V_i$  and  $V_j$  is precisely known. The network localization problem is to determine the unique position of each node in the network given the positions of available beacons and the distance between each pair  $(i, j) \in E$ . If under the given constraints, there is only one position for each node, then the network is localizable. The network localization problem is closely related to the Euclidean graph realization problem, in which coordinates are assigned to vertices of a weighted graph such that the distance between coordinates assigned to nodes joined by an edge is equal to the weight of the edge.

For a two dimensional graph with  $n$  vertices, the positions of its vertices have  $2n$  degrees of freedom, of which three are the rigid body motions. Therefore graph is rigid if there are  $2n - 3$  constraints. If each edge adds an independent constraint, then  $2n - 3$  edges should be required to eliminate all non-rigid motions of the graph. Clearly, if any induced subgraph with  $n$  vertices has more than  $2n - 3$  edges then these edges cannot be independent, which leads to the following Laman theorem [14]:

**Theorem 1** *The edges of a graph  $G = \{V, E\}$  are independent in two dimensions if and only if no subgraph  $G' = \{V', E'\}$  has more than  $2n' - 3$  edges, where  $n'$  is the number of nodes in  $G'$ .*

**Corollary 1** *A graph with  $2n - 3$  edges is generically rigid in two dimensions if and only if no subgraph  $G'$  has more than  $2n' - 3$  edges.*

Laman's theorem characterizes generic rigidity. However, a direct implementation of it leads to a poor exponential algorithm. An efficient approach to check for rigidity is proposed in [11] based on a pebble game. Jacob *et. al* proposed Jacob's approach uses the following formulation of Laman algorithm:

**Theorem 2 [11]** *For a graph  $G = \{V, E\}$  having  $m$  edges and  $n$  vertices, the following are equivalent.*

- *The edges of  $G$  are independent in two dimensions.*
- *For each edge  $(a, b)$  in  $G$ , the graph formed by adding three additional edges identical to  $(a, b)$  has no induced subgraph  $G'$  in which  $m' > 2n'$ .*

The basic idea behind Jacob's algorithm is to grow a maximal set  $S$  of independent edges one at a time. Initially,  $S$  is empty. Let's denote these basis edges by  $E$ . A new edge is added to  $S$  if it is discovered to be independent of the edges existing in  $S$ .

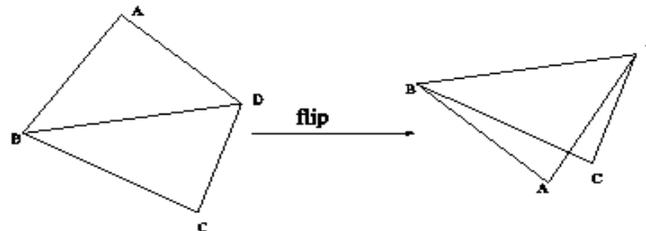


Figure 1: A generically rigid graph subject to flapping transformation. The two realizations are not continuous in two dimension space in that the second one is obtained by a flapping of the first one.

To check whether an edge  $e$  is independent of edges in  $S$ , each vertex is assigned two pebbles initially and a temporary set  $S'$  is created.  $S'$  contains all the edges in  $S$  plus four copies of  $e$ . The pebbles can only travel via the edges in  $S'$ . If all edges in  $S'$  can be covered by the pebbles, then we know that  $e$  is independent of all edges in  $S$  and  $e$  is added into  $S$ . This process is repeated until no more edges can be added into  $S$ . Then  $S$  is a maximal set of independent edges. If  $S$  contains  $2n - 3$  edges, then the graph is generically rigid.

Having  $2n - 3$  independent edges ensures the generic rigidity of a graph. However, it does not guarantee the unique realization of the network. A discontinuous change to the positions of nodes may lead to another realization which satisfies all the constraints of the network, as shown in Figure 1. The following theorem states the condition for a network to be uniquely realizable.

**Theorem 3 [13]** *A graph  $G$  with  $n \geq 4$  vertices is uniquely realizable in two dimensions if and only if it is redundantly rigid and tri-connected.*

Redundant rigidity means after removing any single edge, the remaining graph is still generically rigid. A tri-connected graph is a connected graph such that deleting any two vertices (and incident edges) results in a graph that is still connected. When a network satisfies the condition in Theorem 3 can be uniquely localized given at least three nonlinear beacons in a two dimensional space.

### 3. TRILATERATION TECHNIQUE FOR RIGID GRAPHS

As we have discussed above, a network has to be globally rigid to be localizable. In this section, we are going to discuss in detail the proposed approach. To get a measure of localizability of the network we use a modified trilateration, a well-known technique in localizability. The technique of trilateration is based on the fact that in 2D, the unique location of the node can be determined given the distances to three other nodes whose locations are already known.

In iterative trilateration, we perform trilateration with starting vertices from different geographical parts of the graph, and choose the trilateration graph that has maximum number of localizable nodes.

#### Algorithm 1

1. *Generate a graph that has flip rigidity*
2. *Repeat the following steps up to  $K$  times*
3. *Choose three starting vertices as anchors*
4. *Use trilateration to annex other vertices that connected to Anchor vertices*
5. *Mark the newly annexed vertices as anchors*
6. *Repeat annexation of vertices until no more vertices can be annexed*
7. *Count the number of annexed nodes.*
8. *Go back to step 2 with three other starting vertices.*

## 4. RESULTS OF SIMULATION

We demonstrate that for sparse graphs, there is almost no difference in the performance of trilateration whether the graph is rigid or not. We implement a simulation on Matlab of multiple instances of graphs with 200 nodes over a ground of 100 by 100 with various radii. Figures 2 and 3 show random and rigid network of 200 nodes respectively.

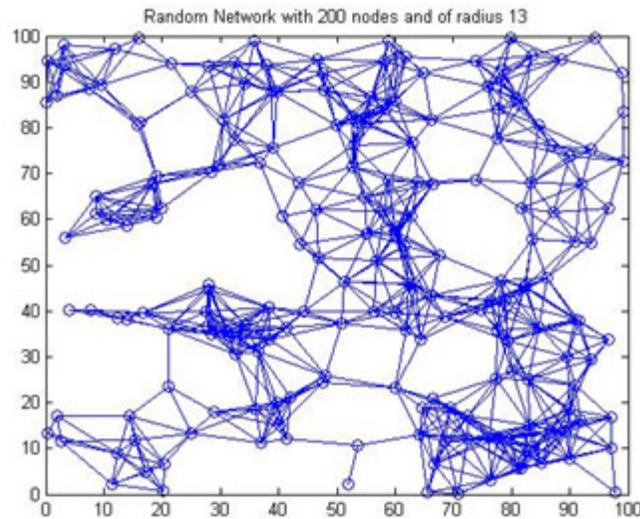


Figure 2: A random network of 200 nodes.

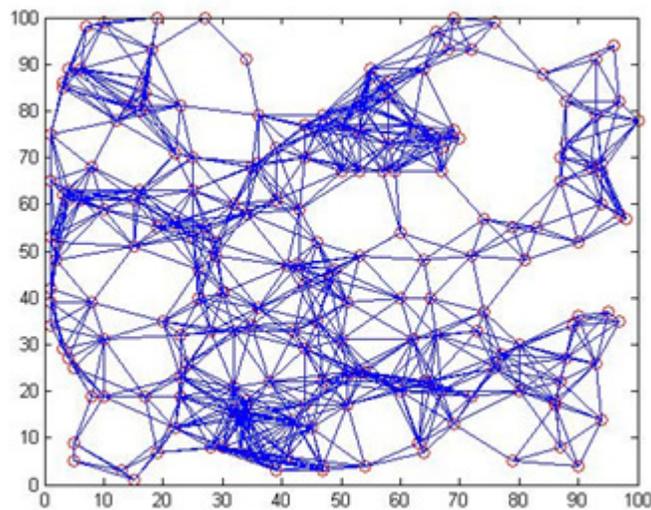


Figure 3: A rigid network of 200 nodes

The simulation is performed for multiple instances of the same radius for both random network and rigid network. We use Pebble game algorithm to check if a graph is rigid. Figure 4 and Figure 5 demonstrate the number of nodes localized for random and rigid networks with iterative trilateration using 100 instances of graphs for the averaging. Figures 6 and 7 show the number of edges used in localization of the corresponding network.

## 5. CONCLUSION

It can be seen that even though rigid networks in general are more localizable, for sparse graphs, only generic rigidity is a possibility and generic rigidity can help only marginally with localization. This is because rigidity is a global property of a graph but trilateration is a local property of a graph. Unless the global property can be translated into a collection of local properties, trilateration is not significantly helped by localization.

This leads us believe that perhaps two different approaches to localization should be considered. For sparse graphs, a localization that is based on finding localizable subgraphs and for dense graphs global rigidity can be used.

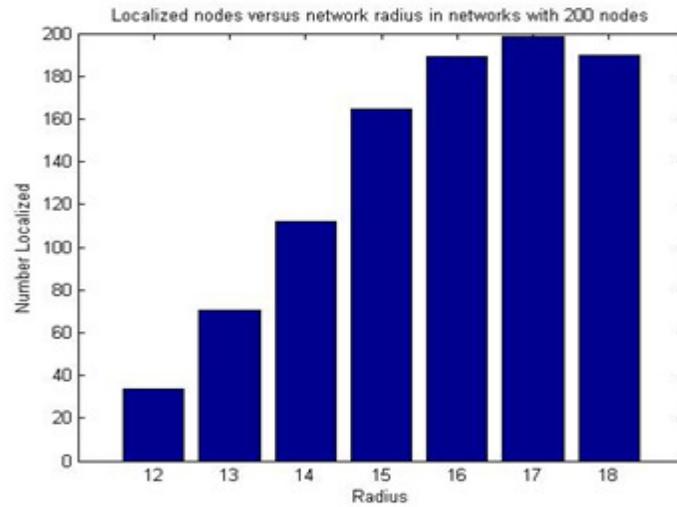


Figure 4: Number of nodes localized in a rigid network of 200 nodes.

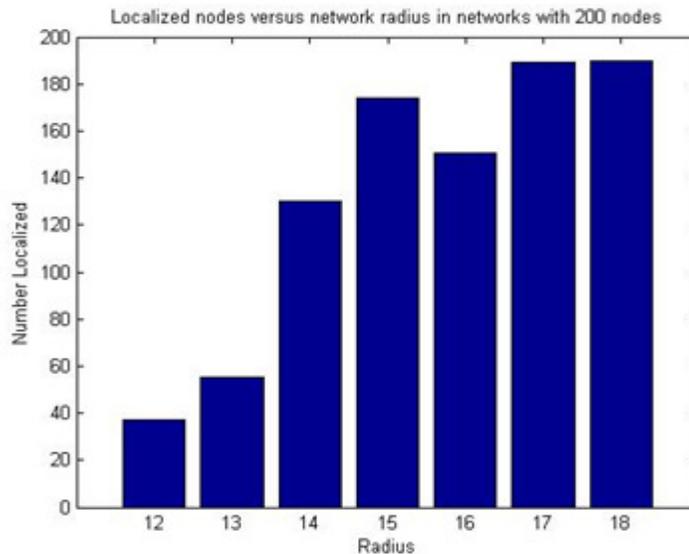


Figure 5: Number of nodes localized in a random network of 200 nodes.

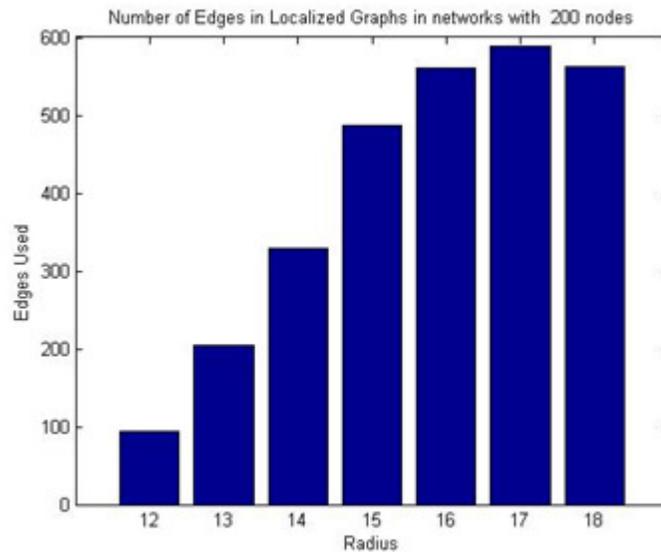


Figure 6: Number of edges used in localization of a rigid network of 200 nodes.

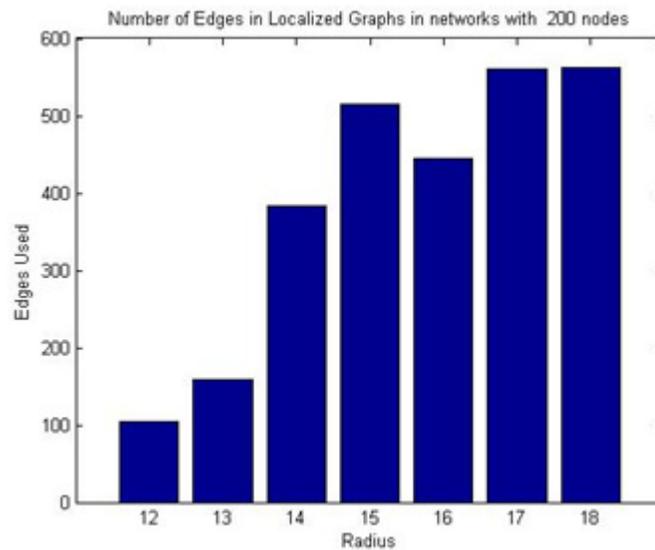


Figure 7: Number of edges used in localization of a rigid network of 200 nodes

## REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Commun. Mag.*, 40:102–114, 2002.
- [2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of Wireless Sensor Network and Applications*, 2002.
- [3] S. N. Simic and S. Sastry. Distributed environmental monitoring using random sensor networks. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, pages 582–592, 2003.
- [4] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Elsevier and Morgan Kaufmann Publishers, 2004.

- [5] C. Savarese, J.M. Rabaey, and J. Beutel, Locationing distributed ad hoc wireless sensor networks. In Proc. 2001 Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001), volume 4, pages 2037–2040, May 2001.
- [6] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. Technical Report 00-729, Computer science department ,University of Southern California, Los Angles, CA, 2000.
- [7] X. Nguyen, M.I. Jordan, and B. Sinopli. A kernel-based learning approach to ad hoc sensor network localization. *ACM Transactions on Sensor Networks*, 1(1):134–152, 2005.
- [8] Zheng Yang and Yunhao Liu. Understanding node localizability of wire- less ad hoc and sensor networks. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, pages 1249–1260, 8 2012.
- [9] Yunhao Liu Zheng Yang and Xiang-Yang Li. Beyond trilateration: On the localizability of wireless ad-hoc networks. In *INFOCOMM*, 2009.
- [10] Isabelle Simplot-Ryl Xu Li, Nathalie Mitton and David Simplot-Ryl. Mobile-beacon assisted sensor localization with dynamic beacon mobility scheduling. 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2011.
- [11] Donald J. Jacobs and Bruce Hendrickson. An algorithm for two-dimensional rigidity percolation: the pebble game. *Journal of Computation Physics*, 137:346–365, 1997.
- [12] T. Eren, O.K. Goldenberg, W. Whiteley, A.S. Yang, Y.R.; Morse, B.D.O. Anderson, and P.N Belhumeur. Rigidity, computation, and randomizationin network localization. In *Proceedings of IEEE INFOCOM 2004, Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2673 – 2684, 2004.
- [13] D. Goldenberg, A. Krishnamurthy, W. Maness, R. Yang, A. Young, and A. Savvides. Network localization in partially localizable networks. In *Proceedings of INFOCOM 2005*, 2005.
- [14] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, pages 331–340, 4 1970.